Amendments to the Specification:

Please insert the following new COPYRIGHT NOTICE paragraph at page 1, line 8, immediately after the paragraph entitled CLAIM TO PRIORITY:

COPYRIGHT NOTICE

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

Please replace paragraph at page 6, lines 4-16 with the following amended paragraph:

Furthermore, VPN services must be very secure. Unfortunately, until rather recently such PC O/S-based VPN support used a rather small key size, such as 40-bits, which generally does not provide sufficient security against third-party intrusion. While relatively new PC-based operating systems have become available that exhibit significantly increased VPN security, through use of triple DES and IPsec features -- such as MICROSOFT® WINDOWS® ~~Microsoft Windows~~ 2000 O/S ("WINDOWS® ~~Windows~~ 2000" is a trademark of the Microsoft Corporation of

Redmond, Washington), this support still presents a considerable processing load to the PC; hence, denigrating PC performance, specifically processing throughput.

Please replace paragraph at page 7, line 1 to page 8, line 2 with the following amended paragraph:

Not only is expensive, specialized VPN equipment required at the client site (or alternatives such as OS-based VPN support or client software packages, both with their accompanying problems, need to be used), but it is also necessary, to an even greater extent, at the LAN (central) site. At the LAN site, VPN support requires installation and configuration of an office-site VPN router. Office-site routers are considerably more expensive than client-site VPN routers for the simple reason that the processing circuitry in the former, which implements the necessary cryptographic and packet processing operations, is sized based on a number of users that need to be simultaneously supported. Each user is allocated a certain slice of the available processing capacity. Even if such a router is sized to support just a few simultaneous remote users on the LAN, its cost can easily amount to several thousands of dollars, with the cost rapidly escalating as user load and hence necessary processing capacity of the VPN router increased. Recently, server operating systems, such as the ~~Microsoft~~ MICROSOFT® WINDOWS® ~~Windows~~ 2000 server O/S, have become available that incorporate multi-user VPN support with sufficient security features; however, such support drains considerable processing resources from the server and still

is insufficiently reliable. Moreover, if such a server O/S-based approach is used, counterpart client-site software, such as the Windows 2000 O/S, must be installed and properly configured on each client PC, which, if a large number of remote users exists, can be rather expensive and time consuming.

Please replace paragraph at page 9, line 17 to page 10, line 7 with the following amended paragraph:

The concept of providing multiple virtual machines is also provided through "Windows Terminal Services" (WTS) software currently available from the Microsoft Corporation ("~~Windows~~WINDOWS® " is a trademark of the Microsoft Corporation) for Windows NT 4 and Windows 2000 server operating systems, with client-server communication of screen shots, keystrokes and mouse clicks being carried to and from WTS using "RDP ('Remote Desktop Protocol' defined by Microsoft Corporation and based on the ANSI T.128 standard)", rather than an "ICA" protocol. Again, WTS, like the Metaframe program, still carries a considerable processing burden.

Please replace paragraph at page 11, line 12 to page 12, line 3 with the following amended paragraph:

To off-load some of the processing burden from the LAN server running WTS, a two-tier approach recently appeared in the art through which a specialized processing system was inserted between the server and a WAN connection. This processor converted RDP packets

associated with WTS into AIP (Application Infrastructure Provider) packets (AIP is a proprietary protocol owned by Tarantella Inc. of Santa Cruz, California) or to some other less bandwidth-intensive protocol and conducted client application communication (screen shots, keystrokes and mouse clicks) with the far-end client PC through either of the latter protocols. ICA provides similar bandwidth conserving functionality. Alternatively, communication in native RDP may be used instead. In any event, the client PC interacted with the processing system through either a specialized client application program or a web browser executing an appropriate JAVA™ applet ("JAVA™" is a trademark of Sun Microsystems, Inc. of Palo Alto, California). While this scheme relieved some of the load on the server, it still suffered the same deficiency as an ASP approach: it was not integrated and thus failed to provide through one single user interface, such as a browser, all the functionality which that user would have if his(her) client PC were directly connected to his(her) office LAN.

Please replace paragraph at page 13, line 26 to page 14, line 11 with the following amended paragraph:

Additionally, network faults can and do occur. However, ~~we~~ it is believed that conventional network management schemes, as conventionally taught in the art, would be rather problematic when used with an office LAN that supports web-based remote user connectivity, and particularly so if network maintenance and management responsibility over that LAN is to be out-sourced, for

reasons of economy -- as would exist in many organizations -- to a third-party vendor. Remote network management, through a centralized location, would be cost-effective provided the scheme chosen to do so could simultaneously monitor and manage a relatively large number of different LANs likely spread over a wide geographic area.

Please replace paragraph at page 19, lines 4-20 with the following amended paragraph:

In accordance with our inventive teachings, the SEP is situated between the LAN and the WAN-connected user. In use, the SEP acts both as a bridge between the remote user and his(her) office applications and as a protocol translator to enable bi-directional, web-based, real-time communication to occur between the user browser and each of these office applications. In that regard, the SEP provides bi-directional protocol translation to exchange necessary information (data and user interactions) between, on the one hand, application-specific protocols, such as MICROSOFT®~~MS~~-RDP, IMAP4 (Internet Mail Access Protocol version 4) or MICROSOFT® .NET technology SMB (Simplified Message Block) ~~MS-Net~~ SMB ~~(Server Message Block)~~, to communicate with office-based client application, e-mail and file servers; and, on the other hand, HTML, in conjunction with HTTP, as required by the user browser or some other protocol, e.g., AIP or the like, used by an applet within the browser.

Please replace paragraph at page 35, line 4 to page 36, line 27 with the following amended paragraph:

In accordance with our inventive teachings and as described in considerable detail below, SEP 200 provides a front end to server 70 for implementing secure, remote, web-based access, through browser 15, by a user situated at client 10 to the network-based office functionality implemented by server 70 and to the same extent as if client PC 10 were directly connected to LAN 65. Server 70 resides on LAN 65 to collectively implement, through separate internal LAN accessible application servers, various office processing applications (tasks) including, through client applications server 72, thin-client hosted application programs; through web-enabled application server 74, remotely-hosted web-enabled thin-client application programs; through e-mail server 76, e-mail messaging; and, through file server 78, shared file access. Each of these servers is conventional, with E-mail server 76 being implemented, for example, by a MICROSOFT® EXCHANGE™ server ~~Microsoft Exchange Server~~ ("MICROSOFT® EXCHANGE™~~Microsoft Exchange~~" is a trademark of Microsoft Corporation of Redmond, Washington). As noted, in small offices, server 70 is typically implemented by a single server computer. Alternatively, rather than using two separate physical computers, server 70 and SEP 200 can be collectively implemented, as indicated by dot-dashed line 60, on one single physical computer -- with the necessary processing needed to implement server 70 being provided by SEP 200. However, to facilitate understanding, we will depict SEP 200, in terms of its functionality,

separate from that of server 70 or any of the hosted applications and servers executing thereon.

Please replace paragraphs at page 39, line 25 to page 41, line 11 with the following amended paragraph:

In doing so, SEP 200 (see FIG. 1) establishes a LAN connection for the remote user that, as far as that user is concerned, places remote client 10 directly on the LAN. By virtue of such a connection, the remote user can, e.g.: (a) send and receive e-mail through server 76 and manipulate his(her) e-mail stored thereon, (b) access, through file server 78, all his(her) files, as well as other shared files, stored on and accessible through LAN 65, (c) remotely execute, through application server 72, any of his(her) thin-client applications hosted thereon, as well as through server 74 remotely execute any of his(her) thin-client web-based applications hosted there, with real-time results of each of these operations being displayed in HTML form on browser 15. Application server 72 receives user mouse clicks and keystroke data and provides user screen shot displays through use of MICROSOFT®~~Microsoft~~ RDP (remote desktop protocol). Web-enabled application server 74 communicates client application information using HTTP. E-mail server 76 utilizes a conventional IMAP4 protocol; while file server 78 communicates user information using MICROSOFT® .NET technology ~~Microsoft MS-NET~~ Simplified Message Block (SMB) data (to implement MICROSOFT®~~Microsoft~~ NET-BIOS functionality). Note, that while SMB and IMAP4 were shown

here as examples, other protocols such as Novell Netware and the POP3 (Post Office Protocol 3) are usable as well.

In essence, as the reader can appreciate, SEP 200 acts both as a bridge between the user and his(her) office applications and as a protocol translator to enable bi-directional, web-based, real-time communication to occur between user browser 15 and each of these office applications. In that regard, the SEP provides bi-directional protocol translation to exchange necessary information (data and user interactions) between, on the one hand, MICROSOFT®~~MS~~-RDP, IMAP4 or MICROSOFT® .NET technology ~~MS-Net~~SMB to communicate with office application servers 72, 76 or 78, respectively; and, on the other hand, HTML and HTTP as required by user browser 15 for non-thin-client applications, and AIP, or a similar protocol, for thin-client control information, or for thin-client user interaction data transfer (i.e., mouse clicks, keystrokes, control data).

Please replace paragraph at page 43, lines 24 to page 44, line 12 with the following amended paragraph:

As shown, SEP 200 contains Ethernet I/F ports 220 and 250 (also referred to as Ethernet ports 1 and 2), V.90 Fax modem 230, and microprocessor 260, all interconnected by local bus 240. Microprocessor 260, which is illustratively any conventional ~~Intel~~ INTEL® ~~Pentium~~ PENTIUM® grade processor or equivalent (having a clock speed of, e.g., 300 MHz or higher) (~~Pentium~~PENTIUM® is a trademark of Intel Corporation of Santa Clara, California),

is itself connected, through bus 267, to memory 270 and via bus 263 to hard drive 280. Memory 270 is illustratively synchronous dynamic random access memory (SDRAM). Hard drive 280 stores program 300 and X.509 certificate 284. During operation of SEP 200, segments of program 300, to the extent needed, are copied from hard drive 280 into memory 270 from which those segments are executed. Memory 270, being volatile, also stores temporary data, as required during program execution.

Please replace paragraph at page 45, line 18 to page 46, line 7 with the following amended paragraph:

Component 305 is formed of a basic O/S kernel 310, illustratively a conventional Linux O/S, with specific additional and conventional Linux modules to implement necessary network and web-based processing, and device operation. These modules include network address translation (NAT) module 320, IP routing module 330, Open SSL module 340, web server 350 (which is currently available under the name "Apache web server"), send mail module 360, TCP/IP processing module 370, PPP (point-to-point) processing module 380 and device drivers 390. Software 300 includes, as its other component, virtual office software 400 that communicates, as symbolized by line 357, through Apache web server 350. Though a Linux O/S kernel is used, O/S 310 could just as easily be implemented using nearly any other PC or server-based operating system, such as a UNIX or MICROSOFT®

WINDOWS® ~~Microsoft Windows~~ operating system -- though, of course, the modules would need to be compatible with the chosen O/S.

Please replace paragraph at page 49, lines 4 to page 50, line 6 with the following amended paragraph:

Virtual office software 400, operative in conjunction with web server module 350, forms a core software component of our present inventive apparatus. In that regard, software 400 (which is discussed in detail below) implements real-time, bi-directional protocol translation, as described above, to enable the user situated at remote PC 10 to remotely control, execute and interact with any office application hosted at server 70 (see FIG. 1). In that regard, through appropriate protocol conversion, software 400, as shown in FIG. 3A, exchanges necessary information (data and user interactions) between, on the one hand, MICROSOFT®~~MS~~-RDP, IMAP4 or MICROSOFT® .NET technology ~~MS-Net~~ SMB to communicate with office application servers 72, 76 or 78 (see FIG. 1), respectively; and, on the other hand, HTML and HTTP (or an intermediate transport protocol, e.g., AIP) as required by user browser 15 for non-thin-client applications, and AIP or a similar protocol for thin-client applications -- all as required to support centralized hosting of office applications (e.g., user application hosting, file serving, and e-mail) but with user interaction and application display occurring remotely at the client computer under the browser. Modules 320, 330, 340, 360, 370 and 380, as shown in FIG. 3A, all provide necessary network packet

processing, including address translation, encryption/decryption and send mail functionality, ancillary to software 400 but necessary to support proper packet communication over a WAN connection between it and both remote PC 10 and individual office applications executing on local server 70 (shown in FIG. 1).

Please replace paragraph at page 52, line 26 to page 53, line 27 with the following amended paragraph:

Incoming packets from the WAN connection, i.e., originating from remote PC 10 (see FIG. 1) and containing user interaction information relevant to thin-client functionality (e.g., starting of a thin-client application, keystrokes and mouse clicks associated with a thin-client application, etc.) flows, as symbolized by dashed line 404a, through Ethernet port 220 (port 1), within the SEP through device drivers module 390, and via the O/S kernel, to TCP/IP processing module 370 for appropriate TCP/IP packet processing, including packet disassembly. From TCP/IP processing module 370, the resulting information in the disassembled packet is provided by O/S kernel 310, to virtual office software 400 via path 404a (paths 404 and 404a are identical except at the very end; path 404 goes to the software 400 via the Apache web server while path 404a goes directly to the virtual office software). Once virtual office software 400 has appropriately processed the information by providing suitable protocol conversion (including performing SSL operations on the data), that information flows directly from software 400 to that office application accessible

through the LAN via path 402, as described below, if necessary (i.e., if it cannot be handled by the virtual office software directly). Information (such as a thin-client screen update for a particular thin-client application, such as MICROSOFT® WORD™, a word processing application software ~~Microsoft Word~~, for example) from the SEP destined to the remote user flows along path 404a and then via path 404 but in an opposite direction to that just described so as to provide the opposite functionality.

Please replace paragraphs at page 10, line 6 to page 60, line 12 with the following amended paragraph:

E-mail application module 430 (described in detail below in conjunction with FIGs. 8-10) interacts with a client e-mail handler (specifically an IMAP client) to access and retrieve user e-mail stored on an e-mail server, such as a MICROSOFT® EXCHANGE™ ~~Microsoft Exchange~~ server, as well as to manipulate stored e-mail residing in the user's e-mail folders (Inbox, Outbox, Sent Mail and the like) on that server. In terms of message reception, module 430 provides a list of received messages, typically with address and truncated content information -- as typically occurs in e-mail clients (such as in MICROSOFT® OUTLOOK™~~Microsoft Outlook~~ e-mail client program; "MICROSOFT® OUTLOOK™~~Microsoft Outlook~~" is a trademark of the Microsoft Corporation of Redmond, Washington) and, once displayed, permits the user to select, and separately and fully display each message, as desired. This module also permits the user to send outgoing e-mail to and through that server.

Thin-client application module 440 (described in detail below in conjunction with FIGs. 11-13) interacts, through the remote desktop protocol (RDP), with a client application program (e.g., MICROSOFT® WORD™, ~~Microsoft Word~~, MICROSOFT® EXCEL™, a spreadsheet software application, ~~Microsoft Excel~~ or other application program; MICROSOFT® WORD™, ~~"Microsoft Word~~" and " MICROSOFT® EXCEL™~~Microsoft Excel~~" are trademarks of the Microsoft Corporation of Redmond, Washington) being hosted on server 70. Module 440 receives user mouse clicks and keystrokes from the user browser, in AIP form, and passes that information, via RDP, to the client application program to control its execution. In return, this module obtains graphical output displays, as screen shots, generated by the client application program and in RDP form, and converts those screen shots into AIP form and then transmits AIP messages, containing the screen shots, back to the user, specifically the user browser for rendering thereat.

Please replace paragraph at page 72, lines 1-15 with the following amended paragraph:

IMAP client component 810 is a conventional e-mail client component that provides rich interaction with a mail server, such as MICROSOFT® EXCHANGE™ ~~Microsoft Exchange~~ server, that supports the IMAP4 protocol. For example, the IMAP client downloads and displays stored e-mail messages, residing on the mail server, from an Inbox associated with a user. The IMAP client then permits the

user to move and copy mail messages from one folder at the server associated with that user (e.g., Inbox) to another such folder (e.g., Sent), as well as delete any such messages from any such folder. E-mail front end 820 is itself formed of state machine 822 and user interaction component 826 which communicate with each other through an application programming interface (API) as symbolized by line 824.

Please replace paragraphs at page 92, line 10 to page 93, line 21 with the following amended paragraph:

If the user then clicks on any such icon, e.g., that associated with the MICROSOFT® WORD™ ~~Microsoft Word~~ program, this interaction being symbolized by line 1330, Java applet 1180 spawns a new browser window (which the applet controls) for use as a user display area for that particular remotely hosted application program. In addition, then user browser 15 provides, as symbolized by line 1335, a "Session_Start" command to thin-client front end 1120. This command includes the name of an application server (server 72 as shown in FIG. 1), appropriate flags, a domain within which the application server runs, password of the user, the name of the application program (here "Word"), a name of a working directory and other related parameters needed to properly and remotely execute the application program (including fully defining its user environment).

In response to the "Session_Start" command, state machine 1200 transitions, as symbolized by line 1215 in FIG. 12A, from null state 1210 to command interpretation state 1220 where it processes this command. Similar to FIGS. 9 and 10, identical circled letters, rather than numerals, are shown in FIGs. 12A, 12B and 13 in order for the reader to visually correlate specific inter-component messages shown in communications 1300 with their corresponding events (including state transitions) in state diagrams 1200 and 1250. Specifically, while in state 1220, the state machine issues, as symbolized by line 1340, an "RDP_CONNECT_REQ" request message to client RDP component 1110 to request a session with a particular client application server. This request contains, e.g., the server, directory, application program (e.g., the MICROSOFT® WORD™ ~~Microsoft Word~~ program) and other information provided, in the Session_Start command, to the thin-client front end. Once this message is issued, state machine 1200 transitions, as indicated by line 1225, to waiting for response state 1230, waiting for a response from the client application server, e.g., server 72 shown in FIG. 1.